



GRADO EN INGENIERÍA DE LA SALUD

Gestión de Fármacos: Base de Datos Farmaceútica

Realizado por
Alejandro Domínguez Recio
David Ramírez Arco
Paula Mariam Lozano Soria
Isabel Nieto Piernagorda
David Vilaseca Pareja

Tutorizado por
Ismael Navas Delgado
José Francisco Aldana Montes

Departamento
Departamento de Lenguaje y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, (Junio 2021)

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
[GRADUADOS EN INGENIERÍA DE LA SALUD]

Gestión de Fármacos: Base de Datos Farmacéutica

Pharmacological Logistics Management

Pharmaceutical Database

Realizado por

Alejandro Domínguez Recio
David Ramírez Arco
Paula Mariam Lozano Soria
Isabel Nieto Piernagorda
David Vilaseca Pareja

Tutorizado por

Ismael Navas Delgado
José Francisco Aldana Montes

Departamento

Departamento de Lenguaje y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2021

Fecha defensa: Junio de 2021

Resumen

Nuestra base de datos consistirá en una gestión de fármacos a aquellos pacientes que poseen una receta tras haber tenido una consulta médica en su centro de salud. Mediante la tarjeta sanitaria, la cual llevará el tratamiento necesario para cada paciente y de esta manera tener un control de fármacos. Además la base de datos contendrá información de las consultas, los diagnósticos, los especialistas y médicos, etc.

Palabras clave:

- Farmacia
- Paciente
- Base de datos
- Fármaco
- Hospital
- Consulta
- Diagnóstico
- Tratamiento
- Funciones
- Procedimientos
- Medicamento
- Prueba
- Médico

Índice

Resumen	3
Índice	4
1. Introducción	5
1.1 Motivación	
1.2 Objetivos	
1.3 Estructura	
2. Descripción del modelo y diseño EER de la base de datos en "papel"	6
3. Diseño e Implementación de la base de datos en Data Modeler	9
3.1 Descripción del proceso de inserción de datos en SQL	12
4. Diseño de consultas Álgebra Relacional y SQL	18
5. Diseño de Procedimientos, Triggers y Funciones	28
6. Desarrollo de aplicación de consola en Python Insertar y consultar datos	46
7. Conclusión	52

1

Introducción

1.1 Motivación

La principal motivación del proyecto es el almacenamiento y disposición efectiva de los datos generados en todo el proceso de un sistema de distribución de fármacos.

1.2 Objetivos

Entre los principales objetivos podemos destacar:

- Un almacenaje efectivo de los datos de todas las entidades participantes en el proceso de distribución de fármacos.
- Disponibilidad de los datos necesarios a la hora de realizar una entrega o solicitud de fármacos.
- Utilización de datos cruzados evitando redundancia en estos.
- Dar soporte a través del almacenamiento de datos en las diferentes situaciones de los procesos.

1.3 Estructura de la memoria

Hemos estructurado la memoria del trabajo guiándonos de las entregas correspondientes de cada semana, tal y como hemos obtenido los conocimientos en clases tanto prácticas como teóricas.

2

Descripción del modelo y diseño EERR

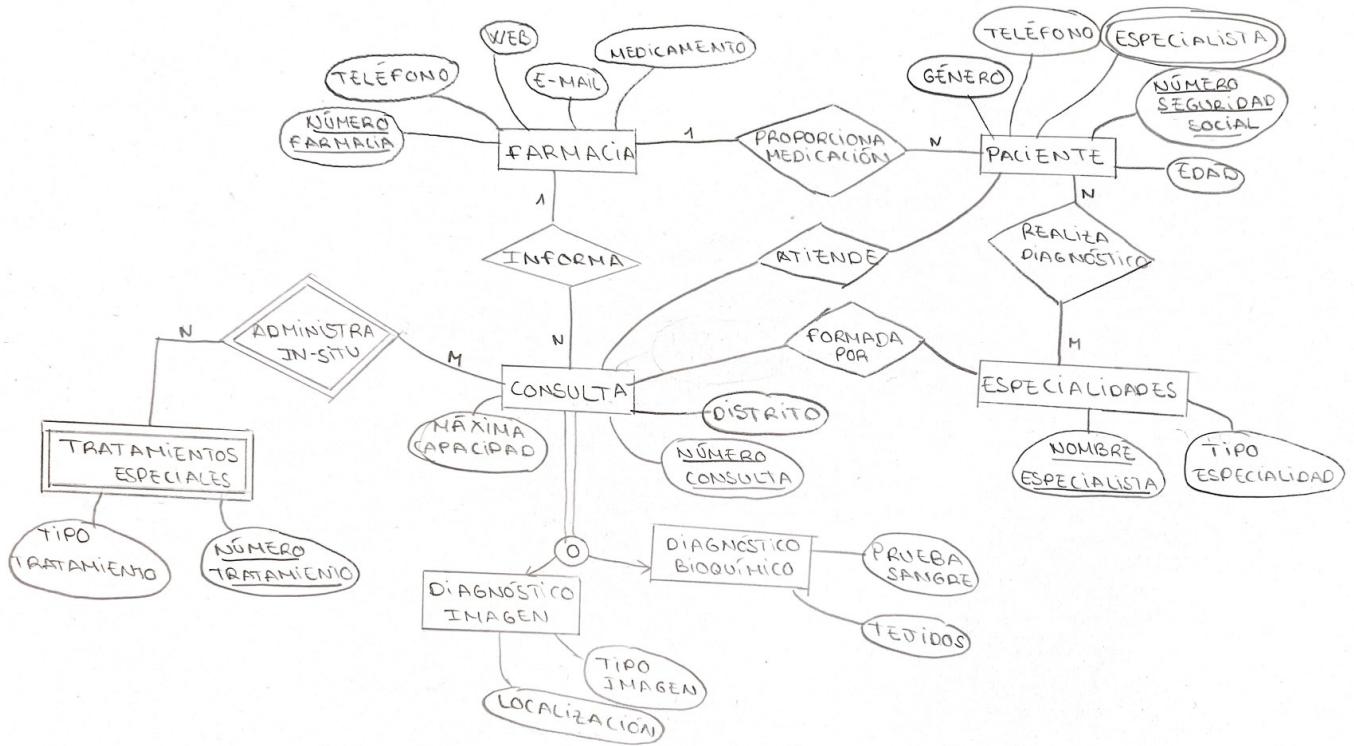
En la realización de nuestra descripción del modelos y diseño EERR recreamos los procesos y los datos generados en estos que se pueden dar en nuestro sistema de distribución de fármacos, los cuales detallamos a continuación:

1. Las farmacias tendrán un teléfono, email, web y un número de consulta identificativo. A la vez mantendrá relación con los pacientes y con las consultas. Cuando un paciente tiene relación con la farmacia está estará identificada con el tipo de medicamento a dispensar siendo posible que una farmacia tenga relación con un número ilimitado de pacientes mientras que un paciente solo puede tener relación con una farmacia.
2. Las consultas se identificarán por un distrito al que pertenecen, la capacidad máxima de pacientes en una consulta y un número identificativo de número de consulta. A la vez, las consultas atenderán a pacientes los cuales podrán asistir a un número ilimitado de consultas como también las consultas podrán atender un número ilimitado de pacientes. Toda consulta debe mantener relación con una farmacia con el fin de facilitar los datos de los tratamientos necesarios. Estas estarán formadas por las diferentes especialidades médicas las cuales estarán identificadas por el tipo de especialidad y su especialista asociado único.

Ciertos tratamientos se realizarán in-situ en la consulta de los cuales se identificarán por el tipo de tratamiento y su número. A la vez cada tratamiento especial que se administre in-situ en la consulta irá identificado en su relación con la fecha de administración de este. No existe restricción de tratamiento que se puedan administrar en una consulta como tampoco el número de consultas que puedan administrar un tratamiento especial.

3. Los pacientes serán identificados por número de la seguridad social único para cada paciente, teléfono, género, edad y especialista, el cual mantiene información sobre su nombre y el número de la seguridad del paciente asociado. Estos mantendrán relación con su farmacia asignada, la consulta en la cual ha sido atendido y la especialidad médica que le esté tratando. El paciente no tendrá límite el número de consultas a las cuales quiera asistir como tampoco el número de especialistas que le puedan dar un diagnóstico.
4. Las consultas tendrán la posibilidad de encargar los servicios de diagnóstico dos tipos de estos, de imagen y bioquímico los cuales irán asociados al número de consulta.
5. Las distintas especialidades médicas mantendrán información sobre el nombre del especialista asignado el cual es único y el tipo de especialidad. Esta mantendrá una relación con los pacientes para la realización de diagnósticos siendo posible que un paciente pueda requerir diagnóstico de un número ilimitado de especialidades médicas y que una especialidad médica pueda realizar diagnóstico de un número ilimitado de pacientes.

Esquema Modelo Entidad Relación



3

Diseño e implementación de la base de datos en Data Modeler

Para la creación de nuestra base de datos vamos a realizar el esquema del modelo lógico en el que hemos creado distintas entidades las cuales poseen relaciones entre ellas.

Primero crearemos la entidad *Farmacia* tendría los atributos:

- **número de farmacia** ▪ teléfono ▪ página web ▪ correo electrónico ▪ medicamento.

La entidad *Paciente* tendría los atributos:

- género ▪ teléfono ▪ especialista ▪ edad ▪ **número de seguridad social**

* El atributo **especialista** lo destacamos como atributo multivaluado ya que dentro de este podemos encontrar más información.

La entidad *Consulta* la crearemos con los atributos de:

- **número de consulta** ▪ capacidad máxima ▪ distrito

La entidad *Diagnóstico de imagen* la crearemos con los atributos de:

- **tipo de imagen** y ▪ localización

La entidad *Diagnóstico bioquímico* la crearemos con los atributos de:

- **prueba de sangre** y ▪ **tejido**.

La entidad *Especialidades* está compuesta por 2 atributos:

- **nombre del especialista** y ▪ **tipo de especialista**.

La entidad *Tratamientos especiales* es una entidad fuerte la cual está formada por:

- **tipo del tratamiento** y el ▪ **número de dicho tratamiento**

Una vez creada las entidades con sus respectivos atributos, añadiremos las siguientes relaciones entre ellas, las cuales pasarán a ser entidades en el modelo relacional.

La entidad farmacia proporciona medicación a los diferentes pacientes que acuden a ella.

Las distintas especialidades realizan el diagnóstico a los pacientes.

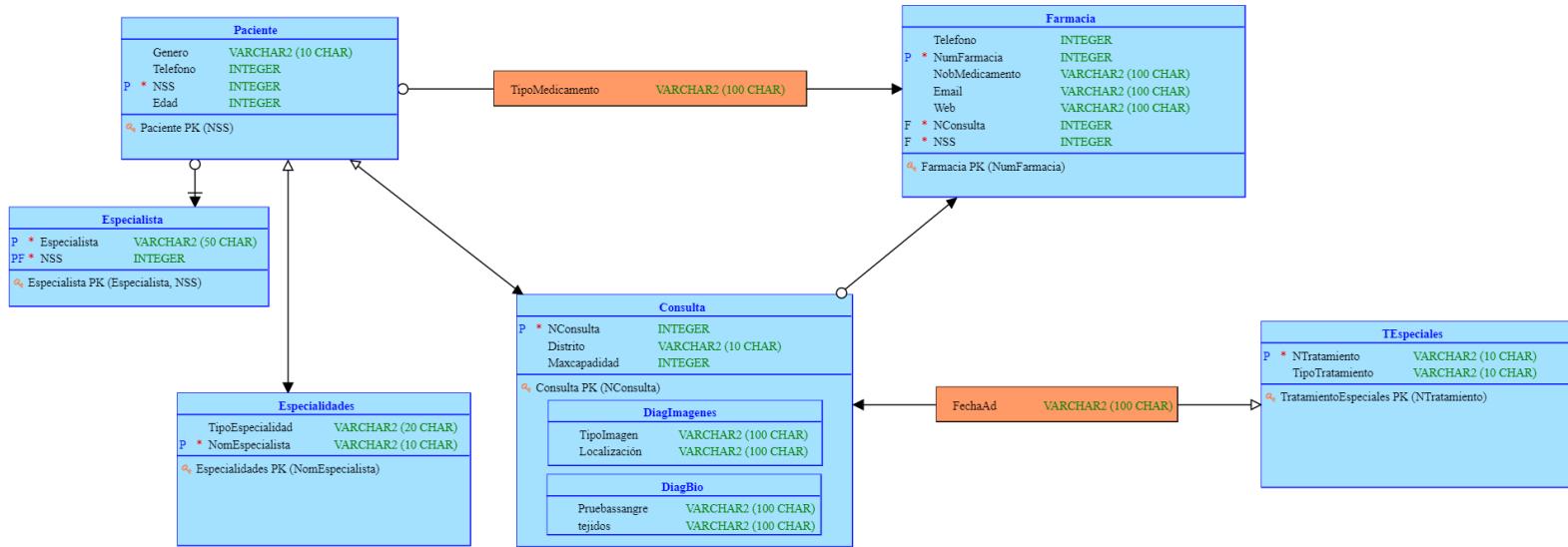
La entidad consulta atiende a un conjunto de pacientes.

La entidad consulta está formada por un conjunto de especialidades según sea los diagnósticos del paciente.

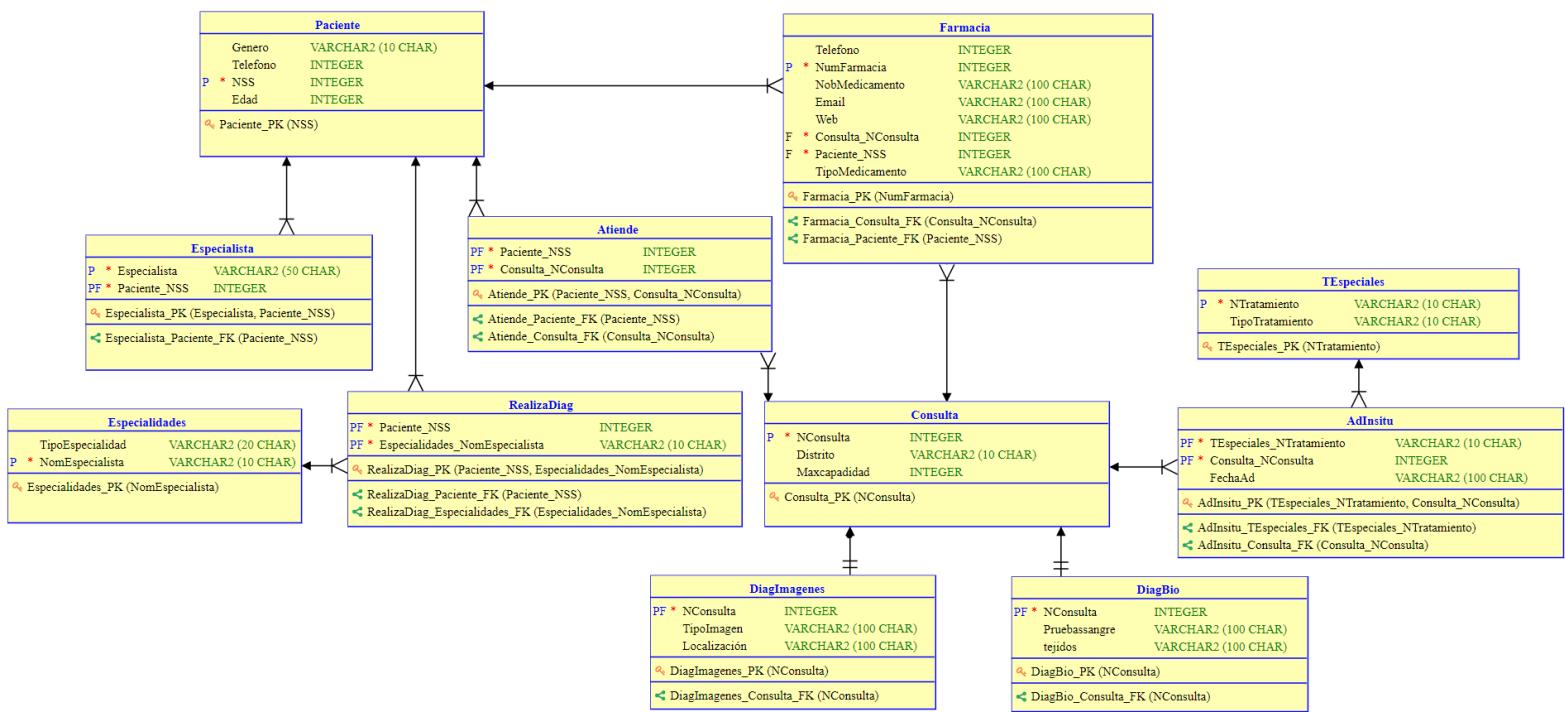
Además la entidad consulta administra in situ los tratamientos especiales que necesitan dichos pacientes.

Con el esquema que hemos diseñado, podemos crear nuestro modelo lógico y relacional, en el que podemos obtener gracias al Data Modeler, las tablas con sus atributos correspondientes y más tarde en SQL Developer se crearía con estas sentencias las bases de datos.

Modelo lógico en data modeler



Modelo relacional en data modeler



3.1 Descripción del proceso de inserción de datos en SQL

En este apartado describiremos el proceso de inserción de datos en la base de datos, para ello crearemos las tablas correspondientes y añadiremos en cada una de ellas los datos necesarios.

La forma de crear estas tablas se diseña directamente a la hora de realizar la ingeniería en data modeler. Data modeller nos proporciona un fichero *.ddl* de la siguiente forma:

```

CREATE TABLE adinsitu (
    tespeciales_ntratamiento VARCHAR2(10 CHAR) NOT NULL,
    consulta_nconsulta INTEGER NOT NULL,
    fechaad VARCHAR2(100 CHAR)
);

```

```

ALTER TABLE adinsitu ADD CONSTRAINT adinsitu_pk PRIMARY KEY
( tespeciales_ntratamiento,
    consulta_nconsulta );

```

```

CREATE TABLE atiende (
    paciente_nss INTEGER NOT NULL,
    consulta_nconsulta INTEGER NOT NULL
);

```

```

ALTER TABLE atiende ADD CONSTRAINT atiende_pk PRIMARY KEY ( paciente_nss,
    consulta_nconsulta );

```

```

CREATE TABLE consulta (
    nconsulta      INTEGER NOT NULL,
    distrito       VARCHAR2(10 CHAR),
    maxcapacidad   INTEGER
);

ALTER TABLE consulta ADD CONSTRAINT consulta_pk PRIMARY KEY ( nconsulta );

CREATE TABLE diagbio (
    nconsulta      INTEGER NOT NULL,
    pruebassangre  VARCHAR2(100 CHAR),
    tejidos        VARCHAR2(100 CHAR)
);

ALTER TABLE diagbio ADD CONSTRAINT diagbio_pk PRIMARY KEY ( nconsulta );

CREATE TABLE diagimagenes (
    nconsulta      INTEGER NOT NULL,
    tipoimagen     VARCHAR2(100 CHAR),
    localización   VARCHAR2(100 CHAR)
);

ALTER TABLE diagimagenes ADD CONSTRAINT diagimagenes_pk PRIMARY KEY ( nconsulta );

CREATE TABLE especialidades (
    tipospecialidad VARCHAR2(20 CHAR),
    nomespecialista VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE especialidades ADD CONSTRAINT especialidades_pk PRIMARY KEY
( nomespecialista );

```

En la entrega final se adjuntará un fichero zip con todos los ficheros completos.

Seguidamente empezaremos insertando en las tablas de la entidad paciente y en ella añadiremos los valores como ‘género’, ‘teléfono’, ‘edad’ y ‘NSS’.

```

INSERT INTO paciente (genero, telefono, nss, edad) VALUES ('Hombre', 952234567, 10, 25);
INSERT INTO paciente (genero, telefono, nss, edad) VALUES ('Hombre', 952123456, 12345678, 25);

```

Seguidamente crearemos la tabla consulta añadiendo de igual forma sus respectivos valores en este caso: ‘nconsulta’, ‘distrito’ y ‘maxcapacidad’.

```

INSERT INTO consulta (nconsulta, distrito, maxcapacidad) VALUES (11, 'Centro', 25);
INSERT INTO consulta (nconsulta, distrito, maxcapacidad) VALUES (123, 'Teatinos', 25);

```

Después, repetiremos este proceso en todas las tablas; añadiremos valores a la tabla especialista como son: ‘especialista’ y ‘paciente_nss’.

```
INSERT INTO especialista (especialista, paciente_nss) VALUES ('Lorenzo Martinez', 12345678);  
INSERT INTO especialista (especialista, paciente_nss) VALUES ('Jose Serrano', 10);
```

En la siguiente tabla, especialidades añadiremos datos como son el ‘tipo de especialidad’ y el nombre de dicha especialidad, ‘nomespecialidad’.

```
INSERT INTO especialidades (tipoespecialidad, nomespecialista) VALUES ('Cardiovascular', 'Lorenzo');  
INSERT INTO especialidades (tipoespecialidad, nomespecialista) VALUES ('Pulmonar', 'Jóse');
```

Posteriormente en la tabla atiende la completamos con sus correspondientes valores: ‘paciente_nss’ y ‘consulta_nconsulta’.

```
INSERT INTO atiende (paciente_nss, consulta_nconsulta) VALUES (10, 11);  
INSERT INTO atiende (paciente_nss, consulta_nconsulta) VALUES (12345678, 123);
```

La siguiente tabla que crearemos será realizadiag, en ella añadiremos: ‘paciente:nss’ y ‘especialidades_nomespecialista’.

```
INSERT INTO realizadiag (paciente_nss, especialidades_nomespecialista) VALUES (12345678, 'Lorenzo');  
INSERT INTO realizadiag (paciente_nss, especialidades_nomespecialista) VALUES (10, 'Jóse');
```

De esta manera seguiremos con todas las tablas necesarias en el siguiente orden: farmacia, diagbio, diagimagenes, tespeciales y adinsitu.

```
INSERT INTO farmacia  
(telefono,numfarmacia,nobmedicamento,email,web,consulta_nconsulta,paciente_nss,tipomedicamento)  
VALUES (952345678,12,'Paracetamol','farmacia12@gmail.com','www.farmacia12.com',11,10,'Analgésico');
```

```
INSERT INTO farmacia  
(telefono,numfarmacia,nobmedicamento,email,web,consulta_nconsulta,paciente_nss,tipomedicamento)  
VALUES (952456789,23,'Ibuprofeno','farmacia23@gmail.com','www.farmacia23.com',123,12345678,'Antiinflamatorio');
```

```
INSERT INTO diagbio (nconsulta, pruebassangre, tejidos) VALUES (11, 'Conteo sanguíneo', 'Piel');  
INSERT INTO diagbio (nconsulta, pruebassangre, tejidos) VALUES (123, 'Colesterol', 'Digestivo');
```

```
INSERT INTO diagimagenes(nconsulta, tipoimagen, localización) VALUES ( 11, 'Radiografía', 'Brazo');  
INSERT INTO diagimagenes (nconsulta, tipoimagen, localización) VALUES ( 123, 'Tomografía', 'Tórax');
```

```
INSERT INTO tespeciales (ntratamiento, tipotratamiento) VALUES ('Ortopedia', 'Pediátrica');
INSERT INTO tespeciales (ntratamiento, tipotratamiento) VALUES ('Logopedia', 'Geriatrífica');
```

```
INSERT INTO adinsitu(tespeciales_ntratamiento, consulta_nconsulta, fechaad)
VALUES ('Ortopedia', 11, TO_DATE('25-04-2021', 'dd-mm-yyyy'));
```

```
INSERT INTO adinsitu (tespeciales_ntratamiento, consulta_nconsulta, fechaad)
VALUES ('Logopedia', 123) TO_DATE (23-05-2021, 'dd-mm-yyyy'));
```

Seguiremos insertando más datos en cada una de las tablas para obtener una base de datos completa. Una vez insertados todos los datos en las tablas, añadiremos las capturas de pantallas correspondientes a dichas tablas para poder observar como han quedado.

The screenshot shows two panes in Oracle SQL Developer. The left pane contains the following SQL code:

```
COMMIT;
select * from adinsitu;
select * from atiende;
select * from consulta;
```

The right pane shows the results of the first query in a table format:

	TESPECIALES_NTRATAMIENTO	CONSULTA_NCONSULTA	FECHAAD
1	Revision		145 29/06/21
2	Cancer		17 23/09/21
3	Logopedia		103 07/07/21
4	Prótesis		53 04/12/21
5	Hidroterapia		18 11/08/21
6	Cirugía		14 25/04/21
7	Ortopedia		11 25/04/21
8	Logopedia		123 23/05/21
9	Quimio		13 03/05/21
10	Fisio		43 23/04/21
11	TMF		19 23/03/21
12	AntiPire		12 23/01/21
13	Stem		12 23/07/21

The right pane also shows the results of the second query in a table format:

	PACIENTE_NSS	CONSULTA_NCONSULTA
1	10	11
2	12115678	13
3	12316442	123
4	12335678	19
5	12345622	12
6	12345678	123
7	12355678	43
8	13466887	12
9	17456678	19
10	27450190	13
11	34786529	18
12	45683127	14
13	53467891	11
14	78451232	17

```
select * from consulta;
```

Salida de Script | Tarea terminada en 0,169 segundos

NCONSULTA	DISTRITO	MAXCAPADIDAD
11	Centro	25
123	Teatinos	25
33	Centro	23
12	La Paz	26
19	Oeste	15
13	Este	25
43	Justicia	25
57	La Luz	27
145	Los prados	10
103	Perchel	13
109	Trinidad	30

NCONSULTA	DISTRITO	MAXCAPADIDAD
73	Portada	16
18	Huelin	23
53	Princesa	25
14	Paraninfo	22
55	Carranque	13
16	La Isla	24
17	Ampliacion	19

18 filas seleccionadas.

```
select * from consulta;
select * from diagbio;
select * from diagimagenes;
select * from especialidades;
```

Salida de Script | Resultado de la Consulta | SQL | Todas las Filas Recuperadas: 12 en 0,196 segundos

NCONSULTA	PRUEBASSANGRE	TEJIDOS
1	18 Linfomas	Inmunologia
2	14 Hematologia	Sangre
3	11 Conteo sanguineo	Piel
4	123 Colesterol	Digestivo
5	13 Transaminasas	Digestivo
6	43 Hemoglobina	Oncologia
7	19 Creatina	Pediatria
8	12 COVID	Respiratorio
9	73 Analisis	Orina
10	57 Anoscopia	Recto
11	33 Artroscopia	Articulacion
12	145 Audiometria	Oido

```
select * from diagimagenes;
select * from especialidades;
```

Salida de Script | Resultado de la Consulta | SQL | Todas las Filas Recuperadas: 12 en 0,095 segundos

NCONSULTA	TIPOIMAGEN	LOCALIZACIÓN
1	18 Análisis	Orina
2	14 Radiografía	Pierna
3	11 Radiografía	Brazo
4	123 Tomografía	Tórax
5	13 Ecografía	Abdomen
6	43 FibroEscan	Higado
7	19 ResonanciaMagnetica	Rodilla
8	12 Tomografía	Cadera
9	145 Ecografía	Vientre
10	33 Ecografía	Espalda
11	57 Nuclear	Pancreas
12	109 ResonanciaMagnetica	Cabeza

`select * from especialidades;`

`select * from especialista;`

TIPOESPECIALIDAD	NOMESPECIALISTA
1 Angiología	Vanesa
2 Urología	Isaac
3 Cirugía	Paco
4 Radiología	Marina
5 Cardiovascular	Lorenzo
6 Pulmonar	José
7 Digestivo	Luisa
8 Pediatría	Ana
9 Oncología	Daniel
10 Endocrinología	Alvaro
11 Traumatología	Maria
12 Neurología	Pedro
13 Radiología	Juan
14 Odontología	Paula
15 Oftalmología	Laura
16 Ginecología	Isabel
17 Hematología	Fernando

ESPECIALISTA	PACIENTE_NSS
1 Alvaro Gonzalez	12115678
2 Ana Lorenzo	12345622
3 Carlos Moreno	45683127
4 Carmen Cobos	78451232
5 Daniel Rovira	123456678
6 Juan Lopez	26450130
7 José Serrano	10
8 José Serrano	12316442
9 José Serrano	17456678
10 Lola Matas	34786529
11 Lorenzo Martinez	12345678
12 Luis Vico	53467891
13 Luisa Mateos	12355678
14 Maria Espina	12345634
15 Maria Espina	13466887
16 Maria Espina	27450190
17 Pedro Perez	12335678

```

select * from farmacia;
select * from paciente;
select * from realizadiag;
select * from tespeciales;

```

`Salida de Script x Resultado de la Consulta x`

`SQL | Todas las Filas Recuperadas: 11 en 0,053 segundos`

TELEFONO	NUMFARMACIA	NOMMEDICAMENTO	EMAIL	WEB	CONSULTA_NCONSULTA	PACIENTE_NSS	TIPOMEDICAMENTO
1 952345678	12 Paracetamol	farmacial2@gmail.com	www.farmacial2.com		11	10	Analgésico
2 952456789	23 Ibuprofeno	farmacia23@gmail.com	www.farmacia23.com		123	12345678	Antiinflamatorio
3 952456789	1 Nolotil	farmacia23@gmail.com	www.farmacia23.com		13	12345622	Analgésico
4 952456789	2 Aspirina	farmacia2@gmail.com	www.farmacia2.com		33	12335678	Analgésico
5 952456789	3 Trombocid	farmacia3@gmail.com	www.farmacia3.com		12	12115678	CremaAntiinflamatoria
6 952456789	4 Prednisona	farmacia4@gmail.com	www.farmacia4.com		11	12355678	Corticoides
7 952456789	13 Abstral	farmacia4@gmail.com	www.farmacial3.com		11	12355678	Antiácidos
8 952456789	9 Bisacodilo	farmacia9@gmail.com	www.farmacia4.com		11	12345678	Laxante
9 952456789	28 Prednisona	farmacia28@gmail.com	www.farmacia28.com		109	27450190	Antifúngico
10 952450789	7 Mucosin	farmacia7@gmail.com	www.farmacia7.com		103	17456678	Mucolítico
11 952250789	25 Sertraline	farmacia25@gmail.com	www.farmacia25.com		145	17456678	Antidepresivos

	select * from paciente;			
	GENERO	TELEFONO	NSS	EDAD
1	Hombre	952987123	45683127	12
2	Mujer	952567324	78451232	33
3	Mujer	952976367	34786529	17
4	Hombre	952136845	53467891	13
5	Mujer	952120971	26450130	16
6	Mujer	952929971	12550190	56
7	Hombre	952234567		10
8	Hombre	952123456	12345678	25
9	Mujer	952123433	12345634	26
10	Mujer	952123444	12345622	22
11	Hombre	952133456	12335678	12
12	Hombre	952113456	12115678	11
13	Mujer	952123556	12355678	11
14	Hombre	9521234456	123456678	15
15	Hombre	9521274155	12316442	9
16	Hombre	9521287611	13466887	48
17	Mujer	9521205212	17456678	56
18	Mujer	9521299718	27450190	36

	select * from paciente;	
	REALIZADIAG	TESPECIALES
1	Revision	Detección
2	Cancer	Quimio
3	Ortopedia	Pediátrica
4	Logopedia	Geriatría
5	Quimio	Oncología
6	Fisio	Trauma
7	TMF	Digestivo
8	AntiPire	Respira
9	Stem	Cardiolgo
10	Cirugia	Ginecolgo
11	Hidroterapia	Trauma
12	Prótesis	Odontologo

4

Diseño de consultas Álgebra Relacional y SQL

En este apartado describiremos el proceso de crear las consultas que desarrollaremos en SQL y en Álgebra Relacional. Al final añadiremos las capturas de pantalla correspondientes a dichas consultas.

- 5 consultas que usen más de dos tablas.
- 3 consultas complejas con el uso de funciones de agregación (anidadas en otras consultas, usando "group by" y usando "having") Resolución de Consultas Complejas.
- 3 consultas complejas con consulta anidada (IN, EXIST, ALL, ANY ...).
- 3 consultas que impliquen la negación de una propiedad existencial Resolución de Consultas Complejas.
- 3 consultas que impliquen una condición "para todo" (cuantificación universal) Resolución de Consultas Complejas.
- 3 consulta usando order by de forma no trivial.

Descripción de las primeras cinco consultas:

- Consulta 1:

Queremos listar el tipo de imagen que se realiza para cada tratamiento especial.

ALGEBRA RELACIONAL

$$\pi_{ADINSITU.NTRATAMIENTO, DIAGIMAGENES.TipoImagen} \sigma_{ADINSITU.Nconsulta = DIAGIMAGENES.Nconsulta} (ADINSITU \times DIAGIMAGENES)$$

SQL

```
select adinsitu.tespeciales_ntratamiento, diagimagenes.tipoimagen from adinsitu,diagimagenes
where adinsitu.consulta_nconsulta = diagimagenes.nconsulta;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 10 en 0,045 segundos

TESPECIALES_NTRATAMIENTO	TIFOIMAGEN
1 AntiPire	Tomoografia
2 Ciruicia	Radioqrafia
3 Fisio	FibroEscan
4 Hidroterapia	Analisis
5 Loquededia	Tomoqrafia
6 Ortopedia	Radioqrafia
7 Quimio	Ecoqrafia
8 Revision	Ecoqrafia
9 Stem	Tomografia
10 TMF	ResonanciaMagnetica

```
select adinsitu.tespeciales_ntratamiento, diagimagenes.tipoimagen from
adinsitu,diagimagenes where adinsitu.consulta_nconsulta =
diagimagenes.nconsulta;
```

- Consulta 2:

En esta consulta listamos el medicamento que se le administra a cada paciente mayor de 10 años, el paciente se identificará por su número de la seguridad social.

ALGEBRA RELACIONAL

$$\pi_{FARMACIA.NOBMEDICAMENTO, PACIENTE.NSS} \sigma_{FARMACIA.NSS = PACIENTE.NSS \text{ and } PACIENTE.EDAD} > 10 (FARMACIA \times PACIENTE)$$

SQL

```
select FARMACIA.NOBMEDICAMENTO, PACIENTE.NSS from FARMACIA,
PACIENTE
where FARMACIA.NSS = PACIENTE.NSS and PACIENTE.EDAD > 10
```

- Consulta 3:

Mostraremos la edad de los pacientes a los que trata cada especialista.

ALGEBRA RELACIONAL

$$\pi_{PACIENTE.EDAD, REALIZADIAG.NOMESPECIALISTA} \sigma_{PACIENTE.NSS = REALIZADIAG.NSS} (PACIENTE \times REALIZADIAG)$$

SQL

```
select PACIENTE.EDAD, REALIZADIAG.NOMESPECIALISTA from
PACIENTE,
REALIZADIAG where PACIENTE.NSS = REALIZADIAG.NSS
```

The screenshot shows a SQL query results window. The query is:

```
select paciente.edad, realizadiag.especialidades_nomespecialista from paciente, realizadiag
where paciente.nss = realizadiag.paciente_nss;
```

The results are displayed in a table with two columns: EDAD and ESPECIALIDADES_NOMESPECIALISTA. The data is as follows:

EDAD	ESPECIALIDADES_NOMESPECIALISTA
1	9 Ana
2	48 Maria
3	56 Pedro
4	36 Daniel
5	25 Jose
6	25 Lorenzo
7	22 Ana
8	12 Maria
9	11 Pedro
10	11 Daniel

The screenshot shows a SQL query results window. The query is:

```
select farmacia.nombmedicamento, paciente.nss from farmacia, paciente
where farmacia.paciente_nss = paciente.nss and paciente.edad > 10;
```

The results are displayed in a table with two columns: NOBMEDICAMENTO and NSS. The data is as follows:

NOBMEDICAMENTO	NSS
1 Mucosin	17456678
2 Sertraline	17456678
3 Prednisona	27450190
4 Paracetamol	10
5 Bisacodilo	12345678
6 Ibuprofeno	12345678
7 Nolotil	12345622
8 Aspirina	12335678
9 Trombocid	12115678
10 Abstral	12355678
11 Prednisona	12355678

Queremos listar el nombre del especialista que hay en cada distrito de la ciudad.

ALGEBRA RELACIONAL

$$\pi \text{ CONSULTA.DISTRITO, REALIZADIAG.NOMESPECIALISTA} \sigma \text{ PACIENTE.NSS} = \text{ATIENDE.NSS} \text{ and } \text{ATIENDE.NCONSULTA} = \text{CONSULTA.NCONSULTA} \text{ and } \text{REALIZADIAG.NSS} = \text{PACIENTE.NSS} \\ (((\text{REALIZADIAG} \times \text{PACIENTE}) \times \text{CONSULTA}) \times \text{ATIENDE})$$

SQL

```
select CONSULTA.DISTRITO,
REALIZADIAG.ESPECIALIDADES_NOMESPECIALISTA from REALIZADIAG,
PACIENTE, CONSULTA, ATIENDE
where PACIENTE.NSS = ATIENDE.PACIENTE_NSS and
      ATIENDE.CONSULTA_NCONSULTA = CONSULTA.NCONSULTA and
      REALIZADIAG.PACIENTE_NSS =
          PACIENTE.NSS
```

```
select consulta.distrito, realizadiag.especialidades_nomespecialista from realizadiag, paciente, consulta, atiende
where paciente.nss = atiende.paciente_nss and atiende.consulta_nconsulta = consulta.nconsulta
and realizadiag.paciente_nss = paciente.nss;
```

DISTRITO	ESPECIALIDADES_NOMESPECIALISTA
1 Centro	José
2 Teatinos	Ana
3 Teatinos	Lorenzo
4 La Paz	Ana
5 La Paz	Maria
6 Oeste	Maria
7 Oeste	Pedro
8 Este	Pedro
9 Justicia	Daniel

- Consulta 5:

Listar los pacientes que sean hombres y se les haya recetado paracetamol.

ALGEBRA RELACIONAL

$$\pi \text{ PACIENTE.NSS, FARMACIA.NOBMEDICAMENTO} \sigma \text{ PACIENTE.GENERO} = \text{'Hombre'} \text{ and } \text{FARMACIA.NSS} = \text{PACIENTE.NSS} \text{ and } \text{FARMACIA.NOBMEDICAMENTO} = \text{'Paracetamol'} \\ ((\text{PACIENTE} \times \text{FARMACIA}))$$

SQL

```

select PACIENTE.NSS, FARMACIA.NOBMEDICAMENTO from PACIENTE,
FARMACIA
    where PACIENTE.GENERO = 'Hombre' and FARMACIA.NSS =
PACIENTE.NSS
    and FARMACIA.NOBMEDICAMENTO = 'Paracetamol'

```

The screenshot shows a MySQL Workbench environment. In the top-left pane, there is a code editor with the following SQL query:

```
--consulta5
select paciente.nss, farmacia.nobmedicamento from paciente, farmacia
where paciente.genero = 'Hombre' and farmacia.paciente_nss = paciente.nss
and farmacia.nobmedicamento = 'Paracetamol';
```

In the bottom-right pane, there is a results grid titled "Resultado de la Consulta". The results are as follows:

	NSS	NOBMEDICAMENTO
1	10	Paracetamol

Descripción de las consultas complejas:

Las siguientes consultas serán complejas y en ellas utilizaremos los comandos 'group by' y 'having':

- Consulta 6: Contar el número de pacientes por género que deben tomar paracetamol.

SQL

```

select paciente.genero, count(paciente.nss) from paciente, farmacia where
farmacia.paciente_nss = paciente.nss and farmacia.nobmedicamento =
'Paracetamol' group by paciente.genero

```

ALGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

The screenshot shows a MySQL Workbench environment. In the top-left pane, there is a code editor with the following SQL query:

```
--consulta6
select paciente.genero, count(paciente.nss) from paciente, farmacia
where farmacia.paciente_nss = paciente.nss
and farmacia.nobmedicamento = 'Paracetamol' group by paciente.genero;
```

In the bottom-right pane, there is a results grid titled "Resultado de la Consulta". The results are as follows:

GENERO	COUNT(PACIENTE.NSS)
Hombre	1

- Consulta 7: Contar el número de pacientes que hay con la misma edad.

SQL

```
select paciente.edad, count(paciente.nss) from paciente group by paciente.edad;
```

ALGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

EDAD	COUNT(PACIENTE.NSS)
1	15
2	36
3	12
4	17
5	56
6	11
7	48
8	33
9	26
10	22
11	9
12	13
13	16
14	25

- Consulta 8: Contar el número de pacientes que hay con la misma edad y que sean mayores de 18 años.

SQL

```
select paciente.edad, count(paciente.nss) from paciente group by paciente.edad having edad > 18;
```

ALGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

EDAD	COUNT(PACIENTE.NSS)
1	36
2	56
3	48
4	33
5	26
6	22
7	25

Posteriormente, consultas en las que utilizaremos 'in', 'all', 'any', 'exist'...:

- Consulta 9: Listar las consultas médicas en las que se administra el tratamiento especial de Logopedia.

SQL

```
select consulta.nconsulta from consulta where consulta.nconsulta in ( select consulta.nconsulta from consulta, adinsitu where adinsitu.tespeciales_ntratamiento = 'Logopedia' and adinsitu.consulta_nconsulta = consulta.nconsulta)
```

ALGEBRA RELACIONAL

No es posible utilizar la función in en álgebra relacional.

The screenshot shows a SQL query being run in a database environment. The query is:

```
-- consulta9
select consulta.nconsulta from consulta where consulta.nconsulta in
( select consulta.nconsulta from consulta, adinsitu
where adinsitu.tespeciales_ntratamiento = 'Logopedia'
and adinsitu.consulta_nconsulta = consulta.nconsulta)
```

The results are displayed in a table titled "NCONSULTA".

NCONSULTA
1
103
2
123

- Consulta 10: Listar las farmacias las cuales su número de identificación sea superior a algunas en las que su número sea 3.

SQL

```
select farmacia.numfarmacia from farmacia where farmacia.numfarmacia >=
any ( select numfarmacia from farmacia where numfarmacia > 3) order by
farmacia
desc;
```

ALGEBRA RELACIONAL

No es posible utilizar la función any en álgebra relacional.

The screenshot shows a SQL query being run in a database environment. The query is:

```
-- consulta10
select farmacia.numfarmacia from farmacia where farmacia.numfarmacia >=
any
( select numfarmacia from farmacia where numfarmacia > 3) order by farmacia.numfarmacia
desc;
```

The results are displayed in a table titled "NUMFARMACIA".

NUMFARMACIA
1
28
2
25
3
23
4
13
5
12
6
9
7
7
8
4

- Consulta 11: Listar los números de consultas médicas que ha realizado diagnóstico por imagen y que se encuentren en la zona este de la ciudad.

SQL

```
select * from diagimagenes where diagimagenes.nconsulta in ( select consulta.nconsulta from consulta where consulta.distrito = 'Este' )
```

ALGEBRA RELACIONAL

No es posible utilizar la función in en álgebra relacional.

NCONSULTA	TIPOIMAGEN	LOCALIZACIÓN
1	13 Ecografía	Abdomen

Además, utilizaremos el comando ‘order by’ para realizar las siguientes consultas:

- Consulta 12: Ordenar las farmacias las cuales su número de identificación (numfarmacia) sea mayor que 3.

SQL

```
select farmacia.numfarmacia from farmacia where farmacia.numfarmacia >= any ( select numfarmacia from farmacia where numfarmacia > 3) order by farmacia.numfarmacia desc;
```

ALGEBRA RELACIONAL

No es posible utilizar la función any en álgebra relacional.

NUMFARMACIA	
1	28
2	25
3	23
4	13
5	12
6	9
7	7
8	4

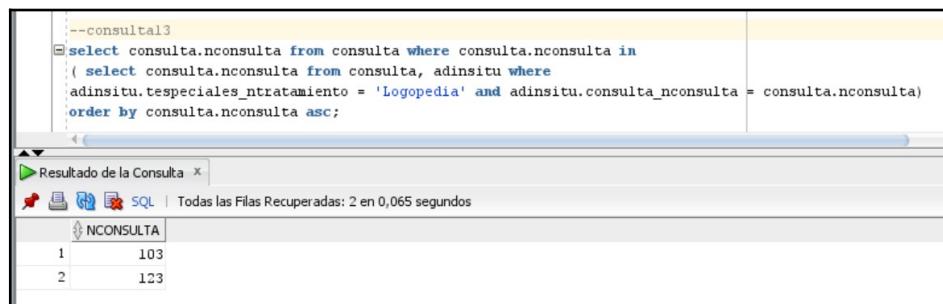
- Consulta 13: Ordenar las consultas médicas en las que el nombre de tratamiento especial sea logopedia.

SQL

```
select consulta.nconsulta from consulta where consulta.nconsulta in ( select
consulta.nconsulta from consulta, adinsitu where
adinsitu.tespeciales_ntratamiento = 'Logopedia' and
adinsitu.consulta_nconsulta = consulta.nconsulta) order by consulta.nconsulta
asc
```

ÁLGEBRA RELACIONAL

No es posible utilizar la función any en álgebra relacional.



The screenshot shows a MySQL Workbench environment. In the top-left pane, there is a code editor containing the following SQL query:

```
--consulta13
select consulta.nconsulta from consulta where consulta.nconsulta in
( select consulta.nconsulta from consulta, adinsitu where
adinsitu.tespeciales_ntratamiento = 'Logopedia' and adinsitu.consulta_nconsulta = consulta.nconsulta)
order by consulta.nconsulta asc;
```

In the bottom-right pane, there is a results grid titled "Resultado de la Consulta". It has one column labeled "NCONSULTA" and two rows of data:

NCONSULTA
1 103
2 123

- Consulta 14: Hacer una lista con el número de pacientes y su edad, además que esta lista esté ordenada de menor a mayor edad.

SQL

```
select paciente.edad, count(paciente.nss) from paciente group by
paciente.edad order by edad asc
```

ÁLGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

The screenshot shows a database management system interface with two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The script tab contains the following SQL code:

```
--consulta 14
select paciente.edad, count(paciente.nss)
from paciente group by paciente.edad order by edad asc
```

The results tab displays a table with two columns: 'EDAD' and 'COUNT(PACIENTE.NSS)'. The data is as follows:

EDAD	COUNT(PACIENTE.NSS)
1	9
2	11
3	12
4	13
5	15
6	16
7	17
8	22
9	25
10	26
11	33
12	36
13	48
14	56

Vamos a realizar las siguientes consultas con el 'not exists':

- Consulta 15: Listar el número de las consultas que han atendido a todos los pacientes.

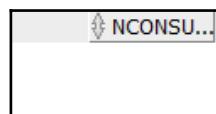
SQL

```
select distinct consulta.nconsulta from consulta where not exists (
  select * from paciente where not exists ( select * from atiende where
  consulta.nconsulta = atiende.consulta_nconsulta and paciente.nss =
  atiende.paciente_nss));
```

ÁLGEBRA RELACIONAL

No es posible utilizar la función not exists en álgebra relacional.

Captura consulta 15:



- Consulta 16: Listar el los nombres de los tratamientos especiales que han sido administrados in situ por todos las consultas.

SQL

```
select distinct tespeciales.ntratamiento from tespeciales where not
exists ( select * from consulta where not exists ( select * from
adinsitu where adinsitu.consulta_nconsulta = consulta.nconsulta and
adinsitu.tespeciales_ntratamiento = tespeciales.ntratamiento));
```

ÁLGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

Captura consulta 16:



- Consulta 17: Listar el número de la seguridad social de los pacientes a los que le han realizado diagnostico todos los especialistas.

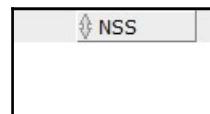
SQL

```
select distinct paciente.nss from paciente where not exists ( select *  
from especialidades where not exists ( select * from realizadiag where  
realizadiag.especialidades_nomespecialista =  
especialidades.nomespecialista and realizadiag.paciente_nss =  
paciente.nss));
```

ÁLGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

Captura consulta 17:



- Consulta 18: Listar todas las consultas que han atendido a mujeres.

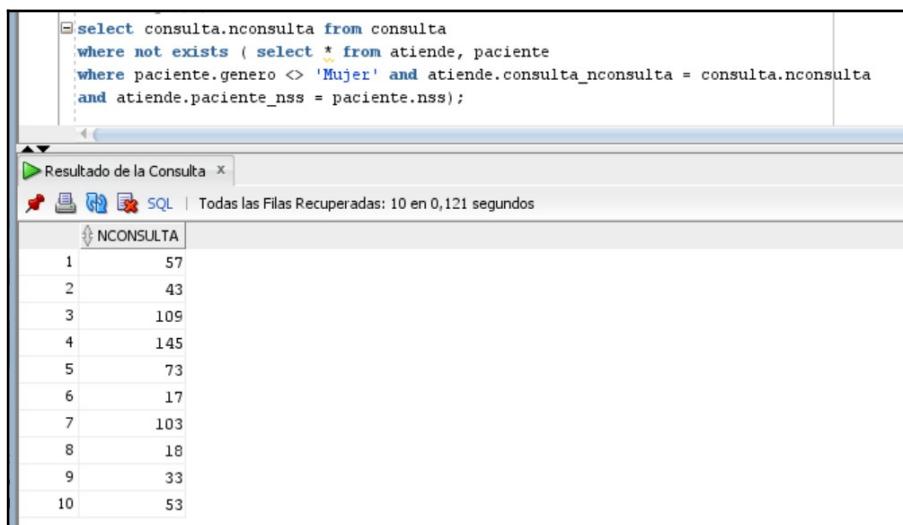
SQL

```
select consulta.nconsulta from consulta where not exists ( select * from
atiende, paciente where paciente.genero <> 'Mujer' and
atiende.consulta_nconsulta = consulta.nconsulta and
atiende.paciente_nss = paciente.nss);
```

ÁLGEBRA RELACIONAL

No es posible utilizar la función count en álgebra relacional.

Captura consulta 18:



The screenshot shows a MySQL Workbench environment. In the top-left pane, there is a SQL editor window containing the following query:

```
select consulta.nconsulta from consulta
where not exists ( select * from atiende, paciente
where paciente.genero <> 'Mujer' and
atiende.consulta_nconsulta = consulta.nconsulta
and atiende.paciente_nss = paciente.nss);
```

In the top-right pane, there is a results grid titled "Resultado de la Consulta". The grid has one column labeled "NCONSULTA" and contains 10 rows of data:

NCONSULTA	
1	57
2	43
3	109
4	145
5	73
6	17
7	103
8	18
9	33
10	53

Below the results grid, a status bar indicates: "Todas las Filas Recuperadas: 10 en 0,121 segundos".

5

Diseño de Procedimientos, Triggers y Funciones

En este apartado vamos a detallar los siguientes procedimientos, triggers y funciones.

Empezaremos con los procedimientos:

1. Procedimiento que obtiene el nombre de los tratamientos administrados in situ y el número de consultas que lo administran.

```
create or replace PROCEDURE PROCEDURE_PROYECTO1
AS
BEGIN
    FOR micursor in (select count(adinsitu.consulta_nconsulta) as Consultas,
        tespeciales_ntratamiento from adinsitu group by
        adinsitu.tespeciales_ntratamiento)
```

```
create or replace PROCEDURE PROCEDURE_PROYECTO1
AS
BEGIN
    FOR micursor in (select count(adinsitu.consulta_nconsulta) as Consultas,
        tespeciales_ntratamiento from adinsitu group by adinsitu.tespeciales_ntratamiento)
    loop
        dbms_output.put_line(micursor.Consultas || ' ' || micursor.tespeciales_ntratamiento);
    end loop;
END PROCEDURE_1
END PROCEDURE_PROYECTO1;
```

Ejecutando: IdeConnections%23Pract+evalu.jpr - Log

Captura procedimiento 1:

```
Conectando a la base de datos Pract evalu.
1 AntiPire
1 Cancer
1 Cirugia
1 Fisio
1 Hidrotera
2 Logopedia
1 Ortopedia
1 Prótesis
1 Quimio
1 Revision
1 Stem
1 TMF
El proceso ha terminado.
```

- Procedimiento que obtiene el tipo de imagen y la localización de esta en los diagnósticos de imagen realizados por una consulta dada.

```
create or replace PROCEDURE PROCEDURE_PROYECTO2
(
    PARAM1 IN diagimagenes.nconsulta%type
) AS
BEGIN
    FOR micursor
        in (select tipoimagen, diagimagenes."LOCALIZACIÓN" from diagimagenes where nconsulta
= PARAM1)
    loop
        dbms_output.put_line(micursor.tipoimagen || ' ' || micursor."LOCALIZACIÓN");
    end loop;
END PROCEDURE_PROJECTO2;
```

Captura procedimiento 2:

```
create or replace PROCEDURE PROCEDURE_PROJECTO2
(
    PARAM1 IN diagimagenes.nconsulta%type
) AS
BEGIN
    FOR micursor
        in (select tipoimagen, diagimagenes."LOCALIZACIÓN" from diagimagenes where nconsulta = PARAM1)
    loop
        dbms_output.put_line(micursor.tipoimagen || ' ' || micursor."LOCALIZACIÓN");
    end loop;
END PROCEDURE_PROJECTO2;

Ejecutando: IdeConnections%23Pract+evalu.jpr - Log
Conectando a la base de datos Pract evalu.
Ecografía Abdomen
El proceso ha terminado.
Desconectando de la base de datos Pract evalu.

Mensajes | Página de Registro | Sentencias | Ejecutando: IdeConnections%23Pract+evalu.jpr | Variables de Salida
```

3. Procedimiento que devuelve el número de consulta y el nss del paciente que toma un medicamento introducido por parámetros.

```
create or replace PROCEDURE PROCEDURE_PROYECTO3
(
    PARAM1 IN farmacia.nobmedicamento%type,
    PARAM2 OUT farmacia.paciente_nss%type,
    PARAM3 OUT farmacia.numfarmacia%type
) AS
BEGIN
    select farmacia.paciente_nss,farmacia.numfarmacia
    into PARAM2, PARAM3
    from farmacia
    where nobmedicamento = PARAM1;
END PROCEDURE_PROYECTO3;
```

Captura procedimiento 3:

```
create or replace PROCEDURE PROCEDURE_PROYECTO3
(
    PARAM1 IN farmacia.nobmedicamento%type
    , PARAM2 OUT farmacia.paciente_nss%type
    , PARAM3 OUT farmacia.numfarmacia%type
) AS
BEGIN
    select farmacia.paciente_nss,farmacia.numfarmacia
    into PARAM2, PARAM3
    from farmacia
    where nobmedicamento = PARAM1;

END PROCEDURE_PROYECTO3;
```

Ejecutando: IdeConnections%23Conexi%C3%B3n1.jpr - Log

Conectando a la base de datos Conexión1.
NSSPaciente = 12335678
NFFarmacia = 2
El proceso ha terminado.
Desconectando de la base de datos Conexión1.

Mensajes | Página de Registro | Sentencias | Ejecutando: IdeConnections%23Conexi%C3%B3n1.jpr | Variables de Salida

Seguidamente las funciones:

1. Queremos implementar una función que nos calcule la edad del paciente más mayor.

```
CREATE OR REPLACE FUNCTION FARMACIA_FUNCION1
RETURN VARCHAR2 AS
valor paciente.edad%TYPE;
mayor_numero paciente.edad%TYPE;
BEGIN
    select max(edad) into mayor_numero from paciente;
    valor :=mayor_numero;
    RETURN (valor);
END FARMACIA_FUNCION1;
```

Captura función 1:

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, the tab for the function 'FARMACIA_FUNCION1' is selected. Below the tabs, there are several icons for navigating between tabs and windows. The main workspace displays the SQL code for creating the function:

```
create or replace FUNCTION FARMACIA_FUNCION1 RETURN VARCHAR2 AS
valor paciente.edad%TYPE;
mayor_numero paciente.edad%TYPE;
BEGIN
    select max(edad) into mayor_numero from paciente;
    valor :=mayor_numero;
    RETURN (valor);
END FARMACIA_FUNCION1;
```

At the bottom of the interface, there is a 'Variables de Salida - Log' window. This window contains two columns: 'Variable' and 'Valor'. The variable '**<Return Value>**' has a value of **26**.

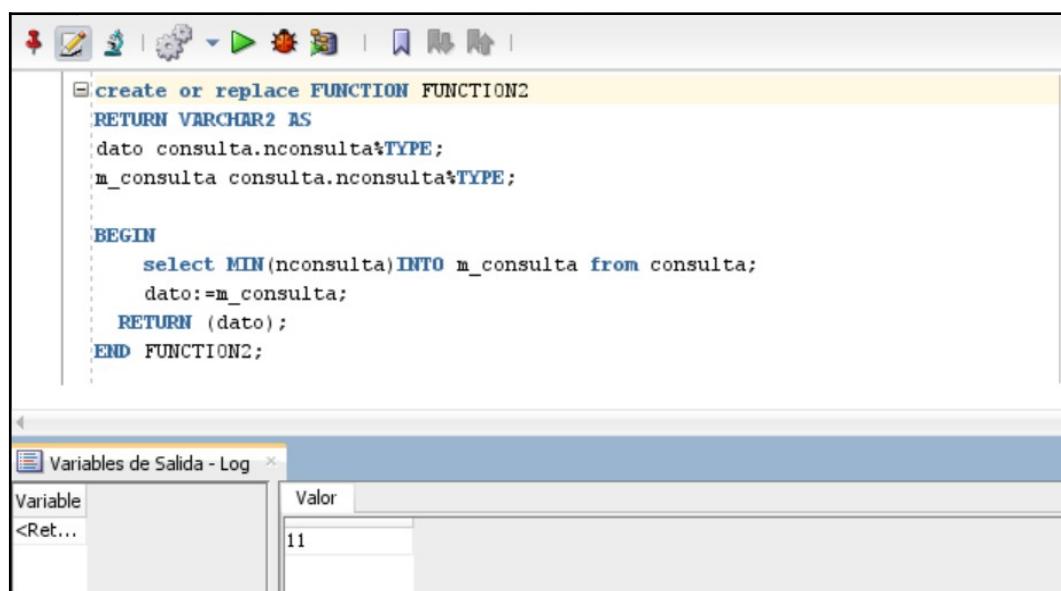
Variable	Valor
<Return Value>	26

2. Hacer una función que encuentre el número de más pequeño de consulta.

```
create or replace FUNCTION FUNCTION2
RETURN VARCHAR2 AS
dato consulta.nconsulta%TYPE;
m_consulta consulta.nconsulta%TYPE;

BEGIN
    select MIN(nconsulta) INTO m_consulta from consulta;
    dato:=m_consulta;
    RETURN (dato);
END FUNCTION2;
```

Captura función 2:



```
create or replace FUNCTION FUNCTION2
RETURN VARCHAR2 AS
dato consulta.nconsulta%TYPE;
m_consulta consulta.nconsulta%TYPE;

BEGIN
    select MIN(nconsulta) INTO m_consulta from consulta;
    dato:=m_consulta;
    RETURN (dato);
END FUNCTION2;
```

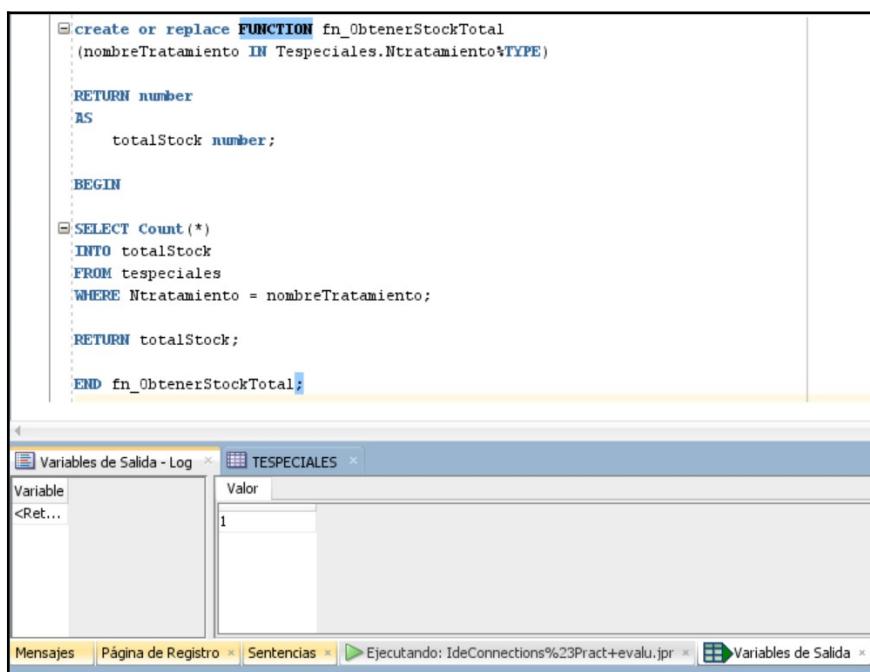
Variable	Valor
<Ret...	11

3. Implementar una función que nos devuelva el stock total de un tratamiento.

```
CREATE OR REPLACE FUNCTION fn_ObtenerStockTotal  
(nombreTratamiento IN Tespeciales.Ntratamiento%TYPE)
```

```
RETURN number  
AS  
    totalStock number;  
BEGIN  
  
SELECT Count(*)  
INTO totalStock  
FROM tespeciales  
WHERE Ntratamiento = nombreTratamiento;  
  
RETURN totalStock;  
  
END fn_ObtenerStockTotal;
```

Captura función 3, con el parámetro = 'Cáncer':



The screenshot shows the Oracle SQL Developer interface. At the top, there is a code editor window containing the PL/SQL function definition. Below it, a 'Variables de Salida - Log' (Output Variables) window is open, showing a single variable entry: 'Variable' <Ret...> and 'Valor' 1. The bottom of the screen displays a navigation bar with tabs: Mensajes, Página de Registro, Sentencias, Ejecutando: IdeConnections%23Pract+evalu.jpr, and Variables de Salida.

```
create or replace FUNCTION fn_ObtenerStockTotal  
(nombreTratamiento IN Tespeciales.Ntratamiento%TYPE)  
  
RETURN number  
AS  
    totalStock number;  
  
BEGIN  
  
SELECT Count(*)  
INTO totalStock  
FROM tespeciales  
WHERE Ntratamiento = nombreTratamiento;  
  
RETURN totalStock;  
  
END fn_ObtenerStockTotal;
```


4. Función que nos devuelva la consulta con la capacidad máxima de un distrito dado.

```
create or replace FUNCTION FUNCTION3_PROYECTO
(
    PARAM1 IN consulta.distrito%type
) RETURN consulta.nconsulta%TYPE AS PARAM2 consulta.nconsulta%type;
BEGIN
    select max(consulta.maxcapacidad) INTO PARAM2 from consulta
    where consulta.distrito = PARAM1;
    RETURN PARAM2;
END FUNCTION3_PROYECTO;
```

Captura función 4, con parámetro 'Este':

```
create or replace FUNCTION FUNCTION3_PROYECTO
(
    PARAM1 IN consulta.distrito%type
) RETURN consulta.nconsulta%TYPE AS PARAM2 consulta.nconsulta%type;
BEGIN
    select max(consulta.maxcapacidad) INTO PARAM2 from consulta where consulta.distrito = PARAM1;
    RETURN PARAM2;
END FUNCTION3_PROYECTO;
```

Variable	Valor
<Ret...	25

5. Función que nos devuelva la edad media de los pacientes de una consulta dada.

```
create or replace FUNCTION FUNCTION1_PROYECTO
(PARAM1 number)
```

```
RETURN NUMBER IS
PARAM2 NUMBER;
BEGIN
  SELECT avg(paciente.edad) into PARAM2 from paciente where
paciente.nss in ( SELECT atiende.paciente_nss FROM ATIENDE WHERE
atiende.consulta_nconsulta = PARAM1 );
  RETURN PARAM2;
END FUNCTION1_PROYECTO;
```

Captura función 5, con numero de consulta = 13 como parámetro:

```
create or replace FUNCTION FUNCTION1_PROYECTO
(PARAM1 number)
  RETURN NUMBER IS
PARAM2 NUMBER;
BEGIN
  SELECT avg(paciente.edad) into PARAM2 from paciente where paciente.nss in ( SELECT atiende.paciente_nss
FROM ATIENDE WHERE atiende.consulta_nconsulta = PARAM1 );
  RETURN PARAM2;
END FUNCTION1_PROYECTO;
```

Variable	Valor
<Ret...	11

6. Implementar una función que nos devuelva la capacidad máxima de una consulta dado su número de consulta y su distrito.

```
create or replace FUNCTION FUNCTION2_PROYECTO
```

```
(PARAM1 consulta.nconsulta%type, PARAM2 consulta.distrito%type)
RETURN consulta.maxcapacidad%type is PARAM3 consulta.maxcapacidad
%type;
BEGIN
  select consulta.maxcapacidad into PARAM3 from consulta where nconsulta
= param1 and distrito = param2;
  return param3;
END FUNCTION2_PROYECTO;
```

Captura función 6, con los parámetros de número de consulta = 11 y distrito 'Centro':

```
create or replace FUNCTION FUNCTION2_PROYECTO
(PARAM1 consulta.nconsulta%type, PARAM2 consulta.distrito%type)
RETURN consulta.maxcapacidad%type is PARAM3 consulta.maxcapacidad%type;
BEGIN
  select consulta.maxcapacidad into PARAM3 from consulta where nconsulta = param1 and distrito = param2;
  return param3;
END FUNCTION2_PROYECTO;
```

Variable	Valor
<Ret...	25

Y por último los triggers:

- Trigger que impide borrar Aspirina del inventario de la farmacia

```
create or replace TRIGGER TRIGGER1_PROYECTO
BEFORE DELETE ON FARMACIA
FOR EACH ROW
BEGIN
    if :old.nombmedicamento = 'Aspirina' then
        RAISE_APPLICATION_ERROR(-20001, 'No se puede borrar Aspirina del inventario.
Medicamento común');
    end if;
END;
```

Captura trigger 1, al intentar borrar Aspirina de la tabla Farmacia:

The screenshot shows a database interface with two main sections. The top section displays a table named 'FARMACIA' with columns: TELEFONO, NUMFARMACIA, NOMBMEDICAMENTO, EMAIL, WEB, CONSULTA_NCONSULTA, PACIENTE_NSS, and TIPOMEDICAMENTO. The data includes various medications like Paracetamol, Ibuprofeno, Nolotil, Aspirina, Trombicid, Prednisona, Abstral, Bisacodilo, Prednisona, Mucosin, and Sertraline, along with their respective details. The bottom section shows a 'Mensajes - Log' window with the following content:

```

Mensajes - Log
Compilado
INSERT INTO "U7675070SP"."ESPECIALISTA" ("ESPECIALISTA", "PACIENTE_NSS") VALUES ('Maria Espina', '11111111')
Confirmación Correcta
Compilado
DELETE FROM "U7675070SP"."FARMACIA" WHERE ROWID = 'AAABLWQAAHAAAAX9AAD' AND ORA_ROWSCN = '57708675' and ( "TELEFONO" is null or "TELEFONO" is not null )
ORA-20001: No se puede borrar Aspirina del inventario. Medicamento común
ORA-06512: en "U7675070SP.TRIGGER1_PROYECTO", linea 3

```

The 'Mensajes' tab is selected at the bottom.

2. Trigger que impide insertar farmacias con número mayor que 23.

```
create or replace TRIGGER TRIGGER2_PROYECTO
BEFORE INSERT ON FARMACIA
FOR EACH ROW
DECLARE numero integer;
BEGIN
    if :NEW.numfarmacia > 23 THEN
        RAISE_APPLICATION_ERROR(-20001, 'No se puede insertar farmacias con número mayor que
23');
    end if;
END;
```

Captura trigger 2, al insertar farmacias con un número mayor a 23:

The screenshot shows a table named 'FARMACIA' with columns: TELEFONO, NUMFARMACIA, NOBMEDICAMENTO, EMAIL, WEB, CONSULTA_NCONSULTA, PACIENTE_NSS, and TIPOMEDICAMENTO. The data includes various medications like Paracetamol, Ibuprofeno, Nolotil, Aspirina, Trombicid, Prednisona, Abstral, Bisacodilo, Prednisona, Mucosin, Sertraline, and Dalsi. The last row (row 12) has values: +12 952456789, 27 Dalsi, far, wwwfar, null, 12, 27450190, Antiinflamatorio. Below the table is a 'Mensajes - Log' window displaying an ORA-20001 error message: 'Se ha producido un error al guardar los cambios en la tabla "U7675070SP"."FARMACIA": Fila 12: ORA-20001: No se puede insertar farmacias con numero mayor que 23'. The bottom status bar shows 'Mensajes Página de Registro Sentencias Ejecutando: IdeConnections%23Pract+evalu.jpr Variables de Salida'.

TELEFONO	NUMFARMACIA	NOMB MEDICAMENTO	EMAIL	WEB	CONSULTA_NCONSULTA	PACIENTE_NSS	TIPO MEDICAMENTO
1 952345678	12 Paracetamol	farmacial2@gmail.com	www.farmacial2.com		11	10	Analgésico
2 952456789	23 Ibuprofeno	farmacia23@gmail.com	www.farmacia23.com		123	12345678	Antiinflamatorio
3 952456789	1 Nolotil	farmacia23@gmail.com	www.farmacia23.com		13	12345622	Analgésico
4 952456789	2 Aspirina	farmacia2@gmail.com	www.farmacia2.com		33	12335678	Analgésico
5 952456789	3 Trombicid	farmacia3@gmail.com	www.farmacia3.com		12	12115678	CremaAntiInflamatoria
6 952456789	4 Prednisona	farmacia4@gmail.com	www.farmacia4.com		11	12355678	Corticoides
7 952456789	13 Abstral	farmacia4@gmail.com	www.farmacia3.com		11	12355678	Antiacidos
8 952456789	9 Bisacodilo	farmacia5@gmail.com	www.farmacia4.com		11	12345678	Laxante
9 952456789	28 Prednisona	farmacia28@gmail.com	www.farmacia28.com		109	27450190	Antifúngico
10 952450789	7 Mucosin	farmacia7@gmail.com	www.farmacia7.com		103	17456678	Mucolítico
11 952250789	25 Sertraline	farmacia25@gmail.com	www.farmacia25.com		145	17456678	Antidepresivos
+12 952478513	27 Dalsi	far	wwwfar		12	27450190	Antiinflamatorio

3. Trigger que impide insertar un tipo de pruebas ya existentes.

```
create or replace TRIGGER TRIGGER3_PROYECTO
AFTER UPDATE ON DIAGBIO
FOR EACH ROW
DECLARE numero INTEGER;
BEGIN
    if :new.pruebassangre = 'Colesterol' then
        raise_application_error(-20000, 'Esta prueba ya está en el registro. Prueba duplicada');
    end if;
END;
```

Captura del trigger 3 al actualizar una fila con 'Colesterol'.

The screenshot shows a database table named 'DIAGBIO' with three columns: 'NCONSULTA', 'PRUEBASSANGRE', and 'TEJIDOS'. The table contains 11 rows of data. Row 5 is highlighted with a blue border. Below the table, a 'Mensajes - Log' window displays the following log entries:

```

UPDATE "U29614238J"."DIAGBIO" SET PRUEBASSANGRE = 'Colesterol' WHERE ROWID = 'AABQQ4AAHAAAABCVAEE' AND ORA_ROWSCN = '57882401'
ORA-20000: Esta prueba ya está en el registro. Prueba duplicada
ORA-06512: en "U29614238J.TRIGGER3_PROYECTO", línea 4
ORA-04088: error durante la ejecución del disparador 'U29614238J.TRIGGER3_PROYECTO'

Se ha producido un error al guardar los cambios en la tabla "U29614238J"."DIAGBIO":
Fila 5: ORA-20000: Esta prueba ya está en el registro. Prueba duplicada
ORA-06512: en "U29614238J.TRIGGER3_PROYECTO", línea 4
ORA-04088: error durante la ejecución del disparador 'U29614238J.TRIGGER3_PROYECTO'

```

- Trigger que impide insertar pacientes con un número de la seguridad social menor a dos dígitos.

```
CREATE OR REPLACE TRIGGER PACIENTEYNSS
BEFORE INSERT ON PACIENTE
FOR EACH ROW
declare numero integer;
BEGIN
if :NEW.nss < 10 THEN
    RAISE_APPLICATION_ERROR(-20002, 'No se puede insertar pacientes con nss menor a 10');
end if;
END;
```

Captura trigger 4, salta el disparador cuando intentamos insertar un número de la seguridad social = 9 (por ejemplo):

The screenshot shows a database interface with a table and a log window.

Table Data:

GENERO	TELEFONO	NSS	EDAD
7 Mujer	952929971	12550190	56
8 Hombre	952234567	10	25
9 Hombre	952123456	12345678	25
10 Mujer	952123433	12345634	26
11 Mujer	952123444	12345622	22
12 Hombre	952133456	12335678	12
13 Hombre	952113456	12115678	11
14 Mujer	952123556	12355678	11
15 Hombre	9521234456	123456678	15
16 Hombre	9521274155	12316442	9
17 Hombre	9521287611	134266887	48
18 Mujer	9521205212	17456678	56
19 Mujer	9521299718	27450190	36
+20 Mujer	952123479	9	48

Log Message:

```
Se ha producido un error al guardar los cambios en la tabla "U7675070SP"."PACIENTE":
Fila 20: ORA-20002: No se puede insertar pacientes con nss menor a 10
ORA-06512: en "U7675070SP.PACIENTEYNSS", linea 4
ORA-04088: error durante la ejecución del disparador 'U7675070SP.PACIENTEYNSS'
ORA-06512: en linea 1
```

Toolbar:

- Mensajes
- Página de Registro
- Sentencias
- Ejecutando: IdeConnections%23Pract+evalu.jpr
- Variables de Salida

5. Trigger que salta al borrar un paciente.

```
CREATE OR REPLACE TRIGGER PACIENTE_BORRADO
BEFORE DELETE ON PACIENTE
FOR EACH ROW
BEGIN
    insert into paciente_borrado (gener,tel,nss2)
    values (:OLD.genero,:OLD.telefono,:OLD.nss);
END;
```

Para este trigger hemos creado la tabla paciente_borrado para que al borrar un paciente se inserte en dicha tabla nueva:

The screenshot shows the Oracle SQL Developer interface. On the left, in the 'Hoja de Trabajo' tab, there is a script editor with the following code:

```
--creo la tabla paciente_borrado
CREATE TABLE paciente_borrado
(gener varchar(10),
tel number(9,0),
nss2 number (10,0) PRIMARY KEY,
edad integer);
```

On the right, in the 'Código' tab, the trigger definition is shown:

```
create or replace TRIGGER PACIENTE_BORRADO
BEFORE DELETE ON PACIENTE
FOR EACH ROW
BEGIN
    insert into paciente_borrado
    (gener,tel,nss2)
    values (:OLD.genero,:OLD.telefono,:OLD.nss);
END;
```

The status bar at the bottom indicates 'Tarea terminada en 0,087 segundos' (Task completed in 0,087 seconds).

The screenshot shows the Oracle SQL Developer interface with a table viewer. The table has columns: GENERO, TELEFONO, NSS, and EDAD. The data is as follows:

GENERO	TELEFONO	NSS	EDAD
1 Mujer	952976367	3445678	17
2 Hombre	952136845	34690345	13
3 Hombre	952127415	12316442	9
4 Hombre	952128761	13466887	48
5 Mujer	952120521	17456678	56
6 Mujer	952129971	27450190	36
7 Mujer	952929971	12550190	56
8 Mujer	952120971	26450130	16
9 Hombre	952234567	10	25
10 Hombre	952123456	12345678	25
11 Mujer	952123433	12345634	26
12 Mujer	952123444	12345622	22
13 Hombre	952133456	12335678	12

At the bottom, a message box displays:

```
DELETE FROM "U51210429D"."PACIENTE" WHERE ROWID = 'AABMBbAAHAAAdpUAAB' AND ORA_ROWSCN = '57920745' and ( "GENERO" is null or "GENERO" is not null )
Confirmación Correcta
```

practicaSQL5.ddl PACIENTE_BORRADO consultasp5.sql BAJAEMPLEADO PACIENTE_BORRADO

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL

GENERO TEL NSS2 EDAD

1 Mujer 952567324 7654321 (null)

Mensajes - Log

Compilado

DELETE FROM "U51210429D"."PACIENTE" WHERE ROWID = 'AABMBbeAAHAAAAdpUAAB' AND ORA_ROWSCN = '57920745' and ("GENERO" is null or "GENERO" is not null)

Confirmación Correcta

6. Trigger que salta cuando una consulta ya reservada se vuelve a reservar en la misma fecha.

```
CREATE or REPLACE TRIGGER ADINSITU_enUso
BEFORE INSERT ON ADINSITU
FOR EACH ROW
DECLARE numero integer;
BEGIN
    if :new.consulta_nconsulta = 14 and :new.fechaad = '25/04/21' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Esta consulta ya esta
reservada para esa fecha');
    end if;
END;
```

Captura trigger 6, salta al añadirse una consulta con la misma fecha:

The screenshot shows a database table named 'ADINSITU' with three columns: 'TESPECIALES_NTRATAMIENTO', 'CONSULTA_NCONSULTA', and 'FECHAAD'. The table contains 14 rows of data. A new row is being inserted with the values 'Hidroterapia', '14', and '25/04/21'. The 'CONSULTA_NCONSULTA' column for this new row is highlighted in green. In the 'Mensajes - Log' window, an error message is displayed: 'ORA-20001: Esta consulta ya esta reservada para esa fecha'. This corresponds to the trigger logic where a reservation for the same date is detected.

TESPECIALES_NTRATAMIENTO	CONSULTA_NCONSULTA	FECHAAD
1 Ortopedia		11 25/04/21
2 Logopedia		123 23/05/21
3 Quimio		13 03/05/21
4 Fisio		43 23/04/21
5 TMF		19 23/03/21
6 AntiPire		12 23/01/21
7 Stem		12 23/07/21
8 Cirugía		14 25/04/21
9 Hidroterapia		18 11/08/21
10 Prótesis		53 04/12/21
11 Logopedia		103 07/07/21
12 Cancer		17 23/09/21
13 Revision		145 29/06/21
+14 Hidroterapia		14 25/04/21

Mensajes - Log

```
INSERT INTO "U26837320T"."ADINSITU" (TESPECIALES_NTRATAMIENTO, CONSULTA_NCONSULTA, FECHAAD) VALUES ('Hidroterapia', '14', '25/04/21')
ORA-20001: Esta consulta ya esta reservada para esa fecha
ORA-06512: en "U26837320T.ADINSITU_ENUSO", linea 4
ORA-04088: error durante la ejecución del disparador 'U26837320T.ADINSITU_ENUSO'
ORA-06512: en linea 1
```

7. Trigger que salta cuando introducimos un paciente cuyo número de teléfono no tiene como prefijo 9.

```
CREATE OR REPLACE TRIGGER TRIGGER8_PROYECTO
BEFORE INSERT ON PACIENTE
FOR EACH ROW
BEGIN
  IF :NEW.telefono not like '9%' then
    raise_application_error(-20000, 'No se admiten teléfonos con prefijos diferentes
de 9');
  end if;
END;
```

Captura del trigger 7 al introducir un paciente con número de teléfono de prefijo diferente de 9.

The screenshot shows a database table named 'PACIENTE' with columns: GENERO, TELEFONO, NSS, and EDAD. The data consists of 18 rows, with the last row being the one being inserted. The 'TELEFONO' column for the last row contains the value '635678976'. A green box highlights this row. Below the table, a 'Mensajes - Log' window displays an error message:

```
ORA-20000: No se admiten teléfonos con prefijos diferentes de 9
ORA-06512: en "U29614238J.TRIGGER7_PROYECTO", linea 3
ORA-04088: error durante la ejecución del disparador 'U29614238J.TRIGGER7_PROYECTO'
ORA-06512: en línea 1
```

At the bottom of the log window, there is additional text:

```
Se ha producido un error al guardar los cambios en la tabla "U29614238J"."PACIENTE":
Fila 19: ORA-20000: No se admiten teléfonos con prefijos diferentes de 9
ORA-06512: en "U29614238J.TRIGGER7_PROYECTO", linea 3
ORA-04088: error durante la ejecución del disparador 'U29614238J.TRIGGER7_PROYECTO'
ORA-06512: en línea 1
```

8. Trigger que salta cuando intentamos añadir un paciente más a un especialista que ya tiene 3 pacientes asociados.

```

create or replace TRIGGER ESPECIALISTAYPACIENTE
BEFORE INSERT ON especialista
FOR EACH ROW
declare
numero integer;
BEGIN
select count(*) into numero
from especialista
where especialista.especialista = :new.especialista;
if numero >= 3 then
raise_application_error(-20000, 'Especialista con más de x pacientes');
end if;

END;

```

Captura trigger 8:

The screenshot shows a database table named 'ESPECIALISTAYPACIENTE' with two columns: 'ESPECIALISTA' and 'PACIENTE_NSS'. The table contains 18 rows of data. A new row is being inserted with the values 'Maria Espina' and '11111111'. The 'Mensajes - Log' tab at the bottom displays an error message indicating that the insertion failed because the specialist already has 3 patients associated with them.

ESPECIALISTA	PACIENTE_NSS
Luis Vico	55407891
Juan Lopez	26450130
Lorenzo Martinez	12345678
Jose Serrano	10
Maria Espina	12345634
Ana Lorenzo	12345622
Pedro Perez	12335678
Alvaro Gonzalez	12115678
Luisa Mateos	12355678
Daniel Rovira	123456678
Jose Serrano	12316442
Jose Serrano	17456678
Maria Espina	13466887
Maria Espina	27450190
Maria Espina	11111111

Mensajes - Log

Se ha producido un error al guardar los cambios en la tabla "U7675070SP"."ESPECIALISTA":
Fila 18: ORA-20000: Especialista con mas de 3 pacientes
ORA-06512: en "U7675070SP.ESPECIALISTAYPACIENTE", linea 8
ORA-04086: error durante la ejecución del disparador 'U7675070SP.ESPECIALISTAYPACIENTE'
ORA-06512: en linea 1

6

Desarrollo de aplicación de consola en Python para insertar y consultar datos

Para el desarrollo de esta parte hemos utilizado:

- ANACONDA gestor de entornos.
- Spyder entorno de desarrollo Python asociado a ANACONDA.
- Librería cx_Oracle de Python la cual nos permite conectar con nuestra base de datos Oracle para así realizar y manipular consultas.

Estructura del código:

- Creamos una función por cada opción de consulta.
- Un bucle while con el que elegir según el dato introducido por parámetro la opción elegida por el usuario.

Consultas predefinidas:

1. Insertar datos en tablas.
2. Insertar datos en relaciones.
3. Consultar datos sobre el número de pacientes que deben de tomar un medicamento dado por parámetro.

4. Consultar datos sobre los diagnósticos de imagen realizados en una consulta de un distrito dado por parámetro.
5. Consultar los números de consulta que realicen un tratamiento especial dado por parámetros.

```

import cx_Oracle

dsnStr = cx_Oracle.makedsn("afrodita.lcc.uma.es","1521","bdsalud")

conexion = cx_Oracle.connect(user="",password="]", dsn=dsnStr)

print(conexion.version)

def consultaInsertparametros(conexion, numeroTabla, values):

    if numeroTabla == 1:

        cursor = conexion.cursor()

        cursor.execute("INSERT INTO paciente (genero, telefono, nss, edad) VALUES ("+values+")")

        cursor.execute("commit")

        cursor.close()

    if numeroTabla == 2:

        cursor = conexion.cursor()

        cursor.execute("INSERT INTO especialista (especialista, paciente_nss) VALUES ("+values+")")

        cursor.execute("commit")

        cursor.close()

    if numeroTabla == 3:

        cursor = conexion.cursor()

        cursor.execute("INSERT INTO especialidades (tipoespecialidad, nomespecialista) VALUES ("+values+")")

        cursor.execute("commit")

        cursor.close()

    if numeroTabla == 4:

        cursor = conexion.cursor()

        cursor.execute("INSERT INTO consulta (nconsulta, distrito, maxcapacidad) VALUES ("+values+")")

        cursor.execute("commit")

        cursor.close()

    if numeroTabla == 5:

        cursor = conexion.cursor()

        cursor.execute("INSERT INTO tespeciales (ntratamiento , tipotratamiento) VALUES ("+values+")")

        cursor.execute("commit")

```

```

cursor.close()

def consultaInsertRelacion(conexion, numeroTabla, values):
    if numeroTabla == 1:
        cursor = conexion.cursor()
        cursor.execute("INSERT INTO atiende (paciente_nss, consulta_nconsulta) VALUES ("+values+")")
        cursor.execute("commit")
        cursor.close()
    if numeroTabla == 2:
        cursor = conexion.cursor()
        cursor.execute("INSERT INTO realizadiag (paciente_nss, especialidades_nomespecialista) VALUES ("+values+")")
        cursor.execute("commit")
        cursor.close()
    def consulta3(conexion, medicamento):
        cursor = conexion.cursor()
        cursor.execute("select paciente.genero, count(paciente.nss) from paciente, farmacia where farmacia.paciente_nss = paciente.nss and farmacia.nobmedicamento = "+medicamento+" group by paciente.genero")
    )
    for resultado in cursor:
        print(resultado)
    cursor.close()

def consulta4(conexion, distrito):
    cursor = conexion.cursor()
    cursor.execute("select * from diagimagenes where diagimagenes.nconsulta in ( select consulta.nconsulta from consulta where consulta.distrito = "+distrito+" )")
    )
    for resultado in cursor:
        print(resultado)
    cursor.close()

def consulta5(conexion, especialidad):
    cursor = conexion.cursor()
    cursor.execute("select consulta.nconsulta from consulta where consulta.nconsulta in ( select consulta.nconsulta from consulta, adinsitu where adinsitu.tepcionales_ntratamiento = "+especialidad+" and adinsitu.consulta_nconsulta = consulta.nconsulta) order by consulta.nconsulta asc")
    for resultado in cursor:
        print(resultado)
    cursor.close()

print("""GESTOR DE BASE DE DATOS"\n")
print("CONSULTAS PREDEFINIDAS DISPONIBLES:")
print(" ")

```

```

print("1.-Insertar datos en tablas")
print("2.-Insertar datos en relaciones")
print("3.-Consultar datos sobre el número de pacientes que deben de tomar un medicamento dado por parámetro")
print("4.-Consultar datos sobre los diagnósticos de imagen realizados en una consulta de un distrito dado por parámetro")
print("5.-Consultar los numero de consulta que realicen un tratamiento especial dado por parámetros")
print("6.-Salir de aplicación ")

print(" ")
print("¿ Cual consulta desea ejecutar ?")

opcion = int(input("Introduzca su elección: "))

while opcion != 6:

    if opcion == 1:

        print("Consulta 1")

        print("Tablas disponibles:\n 1) Paciente \n 2) Especialista \n 3) Especialidades \n 4) Consulta \n 5) TEspeciales \n")

        print("¿ En que tabla desea introducir datos ?")

        opcionTabla = int(input("Número: "))

        if opcionTabla == 1:

            print("Datos necesarios:")

            dato1 = input("Género (Mujer o Hombre) : ")

            dato2 = input("Teléfono: ")

            dato3 = input("NSS: ")

            dato4 = input("Edad: ")

            cadena = ""+dato1+"."+dato2+"."+dato3+"."+dato4

            consultalnsertparametros(conexion,opcionTabla, cadena)

        if opcionTabla == 2:

            print("Datos necesarios: ")

            dato1 = input("Nombre especialista : ")

            dato2 = input("Paciente_nss: ")

            cadena = ""+dato1+"."+dato2

            consultalnsertparametros(conexion, opcionTabla, cadena)

        if opcionTabla == 3:

            print("Datos necesarios: ")

            dato1 = input("Tipo especialidad : ")

            dato2 = input("Nombre especialista: ")

            cadena = ""+dato1+"."+dato1+""

            consultalnsertparametros(conexion, opcionTabla, cadena)

    if opcionTabla == 4:

```

```

print("Datos necesarios: ")

dato1 = input("Número consulta: ")

dato2 = input("Distrito: ")

dato3 = input("Máxima capacidad: ")

cadena = ""+dato1+" "+dato2+" "+dato3

consultalnsertparametros(conexion, opcionTabla, cadena)

if opcionTabla == 5:

    print("Datos necesarios: ")

    dato1 = input("Nombre tratamiento: ")

    dato2 = input("Tipo tratamiento ( Especialidad encargada ): ")

    cadena = ""+dato1+" "+dato1+""

    consultalnsertparametros(conexion, opcionTabla, cadena)

if opcion == 2:

    print("Relaciones disponibles:\n 1) Atiende \n 2) RealizaDiag \n")

    print("¿ En que tabla desea introducir datos ?")

    opcionTabla = int(input("Número: "))

if opcionTabla == 1:

    print("Datos necesarios:")

    dato1 = input("Paciente_nss : ")

    dato2 = input("Número consulta: ")

    cadena = +dato1+" "+dato1

    consultalnsertRelacion(conexion,opcionTabla, cadena)

if opcionTabla == 2:

    print("Datos necesarios: ")

    dato2 = input("Nombre especialista : ")

    dato1 = input("Paciente_nss: ")

    cadena = ""+dato1+" "+dato2

    consultalnsertparametros(conexion, opcionTabla, cadena)

if opcion == 3:

    print("Consulta 3 \n \n ¿ Qué medicamento desea consultar ?")

    opcionMedicamento = input("Introduzca el nombre del medicamento: ")

    consulta3(conexion,opcionMedicamento)

if opcion == 4:

    print("consulta 4 \n \n ¿ Qué distrito desea consultar ?")

    opcionDistrito = input("Introduzca el nombre del distrito: ")

    consulta4(conexion, opcionDistrito)

```

```

if opcion == 5:

    print("Consulta 5 \n \n ¿ Qúe especialidad desea consultar")

    opcionEspecialidad = input("Introduzca el nombre del tratamiento especial: ")

    consulta5(conexion, opcionEspecialidad)

    print("\n CONSULTAS PREDEFINIDAS DISPONIBLES:")

    print(" ")

    print("1.-Insertar datos en tablas")

    print("2.-Insertar datos en relaciones")

    print("3.-Consultar datos sobre el número de pacientes que deben de tomar un medicamento dado por parámetro")

    print("4.-Consultar datos sobre los diagnósticos de imagen realizados en una consulta de un distrito dado por parámetro")

    print("5.-Consultar datos sobre las consultas que realicen un tratamiento especial dado por parámetros")

    print("6.-Salir de aplicación ")

    print(" ")

    print("\n ¿ Cual consulta desea ejecutar ?")

    opcion = int(input("Introduzca su elección: "))

    conexion.close()

```

Captura del menú de inicio de la aplicación:

```

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/aleja/Desktop/Asignaturas/BASES DE DATOS/Prácticas/PartePythonProyecto.py',
               wdir='C:/Users/aleja/Desktop/Asignaturas/BASES DE DATOS/Prácticas')
18.3.0.0.0
'''GESTOR DE BASE DE DATOS'''

CONSULTAS PREDEFINIDAS DISPONIBLES:

1.-Insertar datos en tablas
2.-Insertar datos en relaciones
3.-Consultar datos sobre el numero de pacientes que deben de tomar un medicamento dado por parámetro
4.-Consultar datos sobre los diagnosticos de imagen realizados en una consulta de un distrito dado por
parámetro
5.-Consultar los numero de consulta que realicen un tratamiento especial dado por parámetros
6.-Salir de aplicación

¿ Cual consulta desea ejecutar ?

Introduzca su elección:

```

Captura de acceso al menú de consulta predefinida 1:

CONSULTAS PREDEFINIDAS DISPONIBLES:

- 1.-Insertar datos en tablas
- 2.-Insertar datos en relaciones
- 3.-Consultar datos sobre el numero de pacientes que deben de tomar un medicamento dado por parámetro
- 4.-Consultar datos sobre los diagnosticos de imagen realizados en una consulta de un distrito dado por parámetro
- 5.-Consultar los numero de consulta que realicen un tratamiento especial dado por parámetros
- 6.-Salir de aplicación

¿ Cual consulta desea ejecutar ?

Introduzca su elección: 1

Consulta 1

Tablas disponibles:

- 1) Paciente
- 2) Especialista
- 3) Especialidades
- 4) Consulta
- 5) TEspeciales

¿ En que tabla desea introducir datos ?

Número: |

Captura de menú interno de consulta 1:

```
¿ En que tabla desea introducir datos ?  
Número: 1  
Datos necesarios:  
Género (Mujer o Hombre) : Hombre  
Teléfono: 956345678  
NSS: 23456788  
Edad: 34  
CONSULTAS PREDEFINIDAS DISPONIBLES:  
1.-Insertar datos en tablas  
2.-Insertar datos en relaciones  
3.-Consultar datos sobre el numero de pacientes que deben de tomar un medicamento dado por parámetro  
4.-Consultar datos sobre los diagnosticos de imagen realizados en una consulta de un distrito dado por parámetro  
5.-Consultar datos sobre las consultas que realicen un tratamiento especial dado por parámetros  
6.-Salir de aplicación  
¿ Cual consulta desea ejecutar ?  
Introduzca su elección: |
```

Captura de menú interno y resultado de consulta 4:

```
¿ Cual consulta desea ejecutar ?  
Introduzca su elección: 4  
consulta 4  
¿ Qué distrito desea consultar ?  
Introduzca el nombre del distrito: Este  
(13, 'Ecografía', 'Abdomen')  
CONSULTAS PREDEFINIDAS DISPONIBLES:  
1.-Insertar datos en tablas  
2.-Insertar datos en relaciones  
3.-Consultar datos sobre el numero de pacientes que deben de tomar un medicamento dado por parámetro  
4.-Consultar datos sobre los diagnosticos de imagen realizados en una consulta de un distrito dado por parámetro  
5.-Consultar datos sobre las consultas que realicen un tratamiento especial dado por parámetros  
6.-Salir de aplicación  
¿ Cual consulta desea ejecutar ?  
Introduzca su elección: |
```

*Las capturas han sido elegidas según la estructura de estas en la aplicación.
Diferenciamos dos grupos: grupo uno (C1 y C2) y grupo dos (C3,C4 y C5).

7

Conclusión

Con este apartado finalizamos nuestro proyecto de gestión de fármacos y creación de una Base de Datos Farmacéuticos.

Gracias a este proyecto hemos podido afianzar los conceptos tanto teóricos como prácticos vistos en clase, ha sido de gran ayuda para poder resolver dudas, aprender nociones básicas en aplicaciones que nunca habíamos utilizado antes, como por ejemplo ‘SQL Developer’, incluyendo por lo tanto el desarrollo de nuevos conocimientos.

Además cabe destacar que ha sido de gran interés ver cómo esta idea puede desarrollarse en la actualidad ubicándonos en el mundo laboral.



E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga